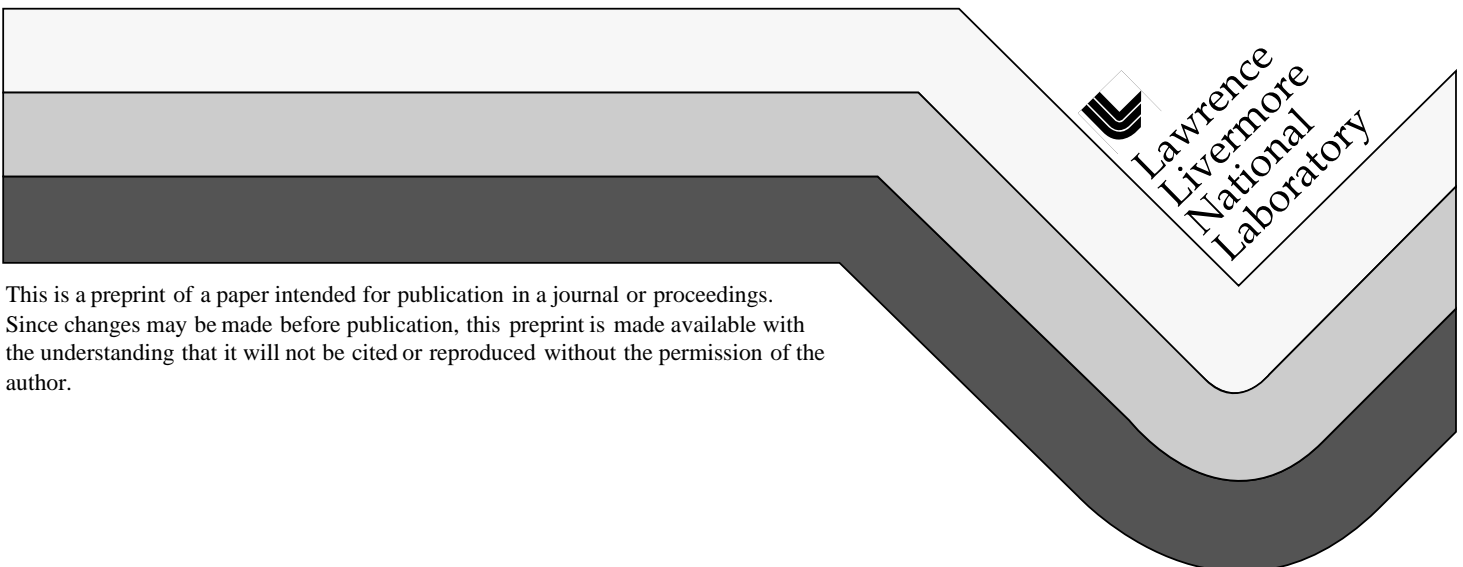


MeshTV: Scientific Visualization and Graphical Analysis Software

L. J. Roberts
E. S. Brugger
S. G. Wookey

This paper was prepared for submittal to the
Nuclear Explosives Code Development Conference
Las Vegas, Nevada
October 25 - 30, 1998

February 8, 1999



DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

MeshTV: Scientific Visualization and Graphical Analysis Software

L. J. Roberts, E. S. Brugger, S. G. Wookey
Lawrence Livermore National Laboratory

The increasing data complexity engendered by the Accelerated Scientific Computing Initiative (ASCI) requires more capability in our scientific visualization software. B Division at Lawrence Livermore National Laboratory (LLNL) addresses these new and changing requirements with MeshTV. We began work on MeshTV around eight years ago, and have progressively refined the software to provide improved scientific analysis and visualization to well over 100 users at Livermore, Los Alamos, Sandia, and in private industry. (U)

Keywords: analysis, graphics, scientific visualization, software, visualization

Introduction

In a post-nuclear test environment, physics simulations become an important window into the nuclear physics world. Of necessity, these simulations have grown in size and complexity in an attempt to model reality. Given that we can no longer test, the accuracy of these simulations becomes ever more crucial as the stockpile ages and we are confronted with situations for which no test data exist.

Scientific analysis and visualization must travel in lockstep with advances in our physics codes. Code developers need to see their data to determine if their codes run correctly. Designers need to see their data to assure their designs. Users need to examine the world of their simulations to develop intuition with new physics models.

As MeshTV developers, we understand the unique nuclear design environment, and we focus on modifying the code to support changes in physics simulations. Our in-house nature gives us the advantage of working alongside the very people tasked to implement the codes of the future. This close communication reflects positively into MeshTV's capabilities.

Code description

MeshTV provides graphical analysis for visualizing and analyzing data on two- and three-dimensional (2D, 3D) finite element meshes. It handles many different mesh types, provides different ways of viewing the data, and removes most kinds of hardware or vendor dependence while still providing graphics at the speed of the native graphics hardware. MeshTV supports collinear and non-collinear quadrilateral meshes, as well as unstructured (UCD) and point meshes. For a definition of these meshes, please see the section titled "Computational mesh descriptions."

MeshTV runs under a Motif-based Graphical User Interface (GUI), or a command line interface. The command line interface can be used either as part of the GUI or as a separate interface which allows inputting direct commands. MeshTV can also read commands from files, allowing users to write MeshTV scripts; to run batch jobs; or to repeat a series of operations on successive files.

Definitions

This document uses some words that might be unfamiliar. Please refer to this section if you encounter unfamiliar terms.

Node

A mathematical point. The fundamental unit of a zone.

Zone	An area or volume forming a mesh-block, or cell. Zones are polygons or polyhedra with nodes as vertices. Zones are occupied by one or more materials.
Block	The fundamental building block of a computational mesh that defines the nodal coordinates of one contiguous section of a mesh, also known as a mesh-block.
Mesh	A collection of one or more blocks. A mesh can be composed of blocks of different types, such as quadrilateral or unstructured. Definitions of the various types of meshes referred to in this document are included in the Computational mesh descriptions section.
Material	A physical material being modeled in a computer simulation. A material consists of one or more material species.
Material Species	A single component of a material. A material contains one or more material species. For example, the material Air contains the species Oxygen and the species Nitrogen. The portion of the material in a zone for a particular material species is measured by mass fraction, since the species are assumed to be distributed evenly throughout the material in the zone.
Multi-Species Material	A material with more than one material species.
Single-Species Material	A material with one material species.
Mixed Zone	A zone containing multiple materials. Also called a Mixed Material Zone.
Clean Zone	A zone containing only one material.
Variable	Data associated with a computational mesh. Variables usually represent values of some physics quantity, like pressure or density. Values are located either at the mesh nodes or as constants throughout the zones.

Capabilities

Plot types. The plot types MeshTV implements include:

- Material boundary plots (lines or filled areas for 2D mesh data, surfaces for 3D meshes).
- Iso-contours (lines or pseudocolor for 2D mesh data, surfaces for 3D meshes).
- Block plots (domain decompositions).
- Mesh plots.
- Label plots (node numbers, zone numbers, or values of a variable).
- Vector plots.
- 3D surface plots of 2D mesh data.

- Stereoscopic viewing of any 3D image on machines with stereoscopic support.

Operations. Often scientists want to visualize the results of operations on their simulation data. MeshTV provides the following operations:

- Reflection.
- Orthogonal and arbitrary slicing of 3D data.
- Material species selection.
- Index selection (i.e., zoom in on a region that is specified by nodal indices).
- Algebraic and mathematical function manipulation of data. MeshTV plots the results of expressions involving arithmetic operators and mesh variables operands.

Comparison of Data. MeshTV allows comparison of data through the following features:

- Multiple graphics windows for side-by-side comparison.
- Overlaying of images.
- Simultaneous display of data from many different files (or different data from the same file) either in separate windows or as overlays.

Data query. MeshTV provides quantitative data to the user. The user obtains these data through the following techniques:

- Pick and query in 2D. When the user picks a point on the plot with the mouse, MeshTV prints the coordinates and the value of the variable plotted (such as pressure, temperature, or density) to a separate window.
- Lineouts, or value versus distance plots. When the user draws a line across the plot, MeshTV plots the value of the variable as a function of distance along the line to a separate window.
- Labels. Users can create a label plot of a variable. This places the value of the variable at the corresponding node or zone.

Parallel code. MeshTV runs in parallel on various parallel platforms. This new functionality will doubtless suffer some problems, but MeshTV's developers are dedicated to fully supporting this capability.

Animation. MeshTV can save a series of raster images, then replay them to produce an animation on the screen. MeshTV can also automatically generate MPEG movies, though this capability is new and is currently very limited.

Distributed mode. MeshTV can operate in a distributed mode. A running simulation on a remote machine can interactively send data directly to MeshTV. The user can interact with this data in the same manner as with data from a file. Many simulations on different machines can send data to a single MeshTV executable running on the user's workstation. This data is displayed in different windows.

File formats. MeshTV can read two different file formats: the graphics file produced by DYNA3D (which uses IEEE 32-bit floating-point numbers) and the Silo format. Silo implements an application program interface expressly designed for accessing scientific data. The Silo library defines a set of objects for handling different meshes. Data in a Silo file is self-describing (name, array rank, dimension size, etc.) This allows MeshTV to display the names of the variables in the selected file as menu selections in the GUI, so the user knows the contents of the file and what can be plotted. We plan to rework Silo to sit on top of HDF5, when it is released.

Future capabilities. We consistently update MeshTV with new capabilities. Some planned extensions include:

- Time–history plots.
- Pick and query capability in 3D.
- Interactive 3D selection of slice planes.
- “Onion-peel” capability. This feature would allow users to select a zone and then the zone’s neighboring zones. The user can then show the next layer of neighboring zones, and so on.

Computational mesh descriptions

Quadrilateral-based meshes and related data. A quadrilateral mesh contains four nodes per zone in 2D and eight nodes per zone (four nodes per zone face) in 3D. Quadrilateral meshes can be either collinear or non-collinear, but they must be logically rectangular. Some users refer to collinear meshes as rectilinear meshes, and to non-collinear meshes as curvilinear meshes. See Figure 1 for an example of the two mesh types.

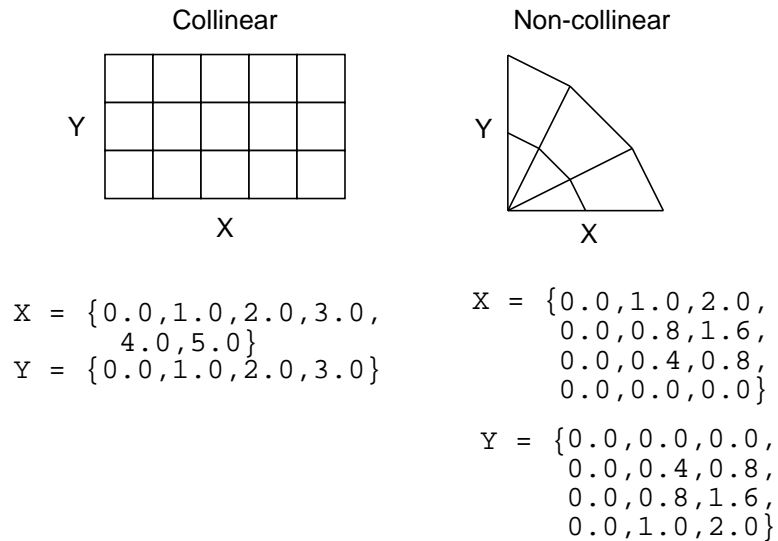


Figure 1. Examples of quadrilateral meshes

A quadrilateral mesh can have “phony” zones—a layer of zones adjacent to one or more of the “real” mesh boundaries (Figure 2). This feature allows codes to simulate boundary conditions,

such as a pressure profile. These zones contain data used in physics simulations for calculations in nearby zones, but graphics applications should never plot them.

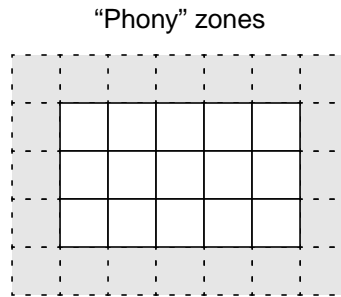


Figure 2. Phony zones around a collinear quadrilateral mesh.

Unstructured meshes and related data. An unstructured mesh is a general mesh representation composed of an arbitrary list of zones of arbitrary sizes and shapes. An unstructured mesh can represent most meshes, including quadrilateral. (Figure 3). However, because of their generality, unstructured meshes require more storage space and algorithms with greater complexity to generate scenes, such as a 2D slice from a 3D data set.

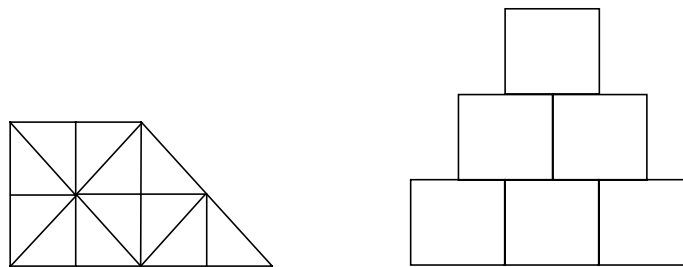


Figure 3. Sample 2D unstructured meshes

In unstructured meshes, the basic concept of zones (cells) still applies, but, unlike the quadrilateral mesh, there is no longer an implied connectivity between a zone and its neighbor. For example, given a 2D quadrilateral mesh zone accessed by (i, j) , you can calculate its neighbors as $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, and so on. There is no way to calculate the neighboring zones on an unstructured mesh.

In an unstructured mesh, a list of nodes defines each zone. An unstructured mesh might be composed of zones of different shapes. Figure 4 shows the zone shapes B Division currently uses.

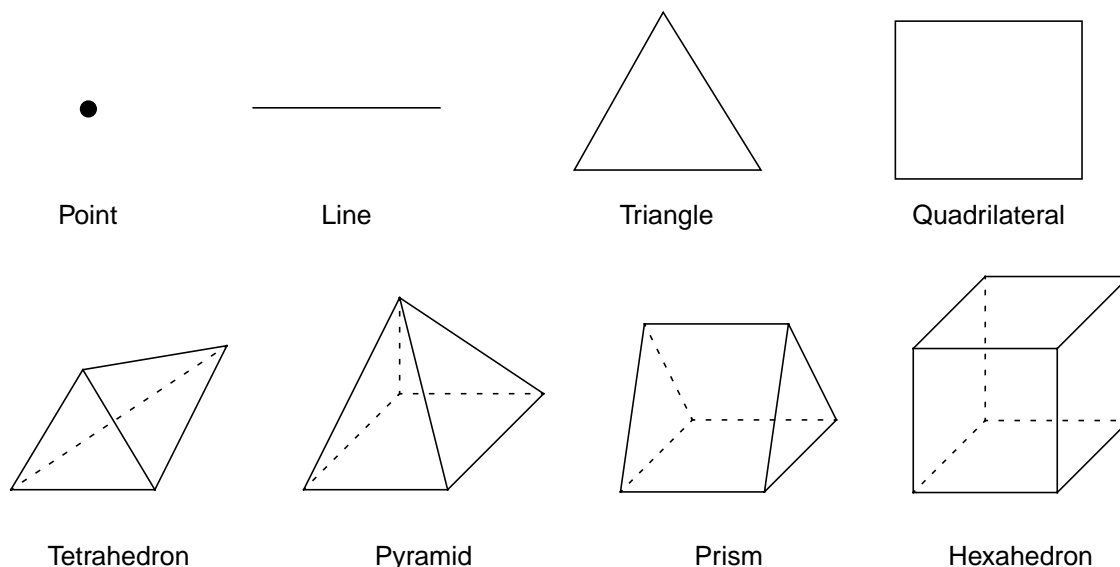


Figure 4. Unstructured 2D and 3D cell shapes

Point meshes and related data. A point mesh consists of a set of locations, or points, in space. Physics simulations use this type of mesh to represent random scalar data, such as tracer particles.

Contact info and documentation

MeshTV's web site (<http://www.llnl.gov/bdiv/meshtv/>) provides free source code, documentation, and examples of code capabilities. You can download executables for the following operating systems:

- IBM AIX
- HP_UX 9
- Irix5 (SGI)
- Irix6 (SGI)
- Linux
- OSF
- Solaris (Sun)

MeshTV documentation exists in both PostScript and HTML formats. Documentation includes the *MeshTV Getting Started Manual*, which walks the user through a tutorial, the *MeshTV Command Line Interface Manual*, which details the command syntax for using MeshTV in batch mode, and the *MeshTV User's Manual*, which describes MeshTV's Graphical User Interface. (The *User's Manual* should be released by January 1999.) The accompanying *Silo User's*

Guide helps code developers instrument their codes to produce Silo files, either restart files or graphics data.

You can email MeshTV developers at MeshTV@llnl.gov. We would love for you to use MeshTV, and we're happy to help get you started.

References

Ahern, S. D., *Silo User's Guide*, Lawrence Livermore National Laboratory, Livermore, CA, UCRL-MA-118751 (1996).

Roberts, L. J., and Brugger, E. S., *MeshTV Getting Started Manual*, Lawrence Livermore National Laboratory, Livermore, CA, UCRL-MA-127442 (1997).

Roberts, L. J., and Brugger, E. S., *MeshTV Command Line Interface Manual*, Lawrence Livermore National Laboratory, Livermore, CA, UCRL-MA-127443 (1997).

Roberts, L. J., *MeshTV User's Manual*, Lawrence Livermore National Laboratory, Livermore, CA, To be released.

Auspices

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng.-48.